Microservice Architecture

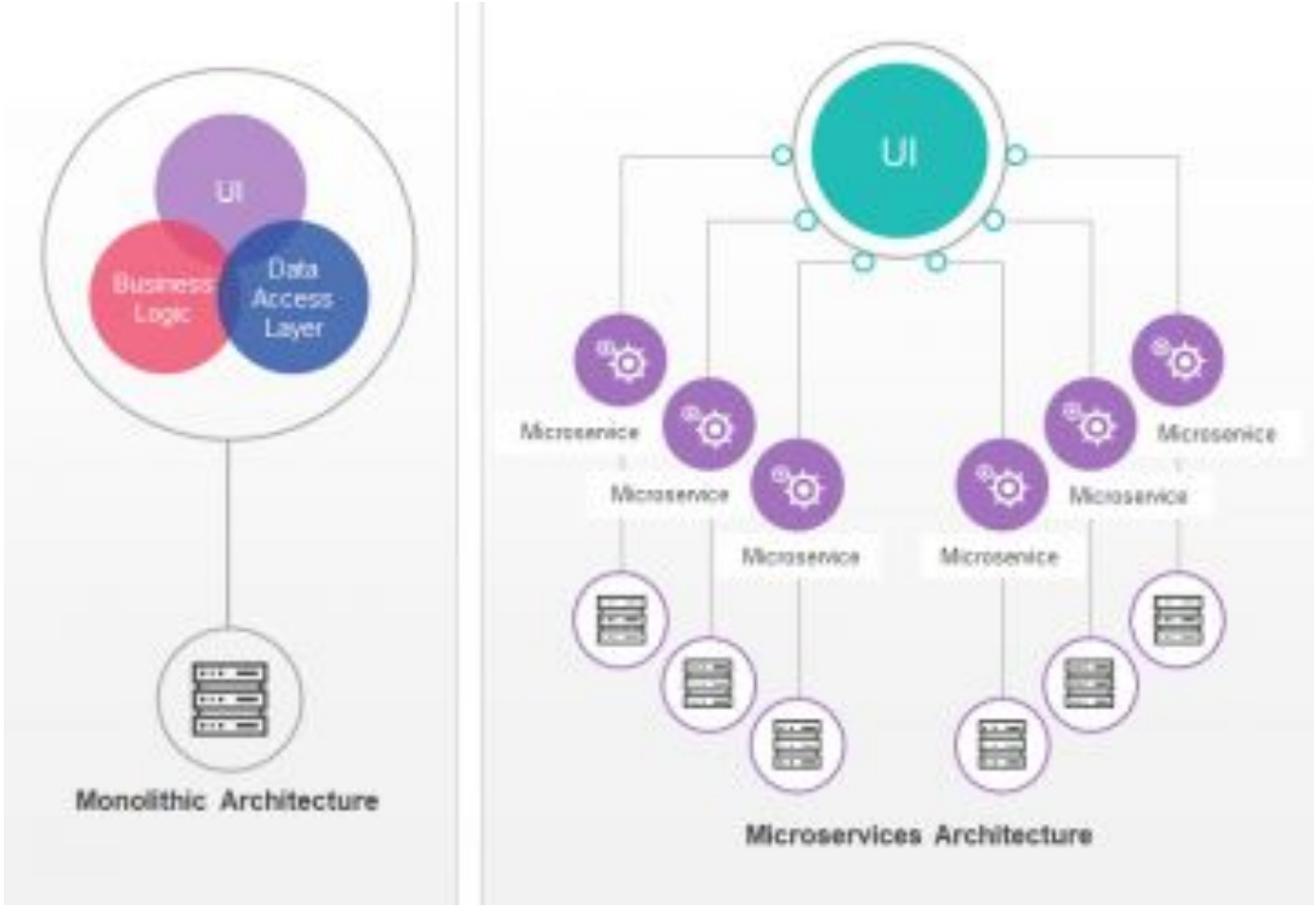
Monolith Application

- Server side system based on single application.
- Easy to manage (?)
- Highly dependent
- Same language / database / framework
- Scalability - Vertical not horizontal


Budget
Field
Production
Materials
Sales
Logistics
Finance
Payroll
MIS
PanAgro
Next Generation Plantation ERP

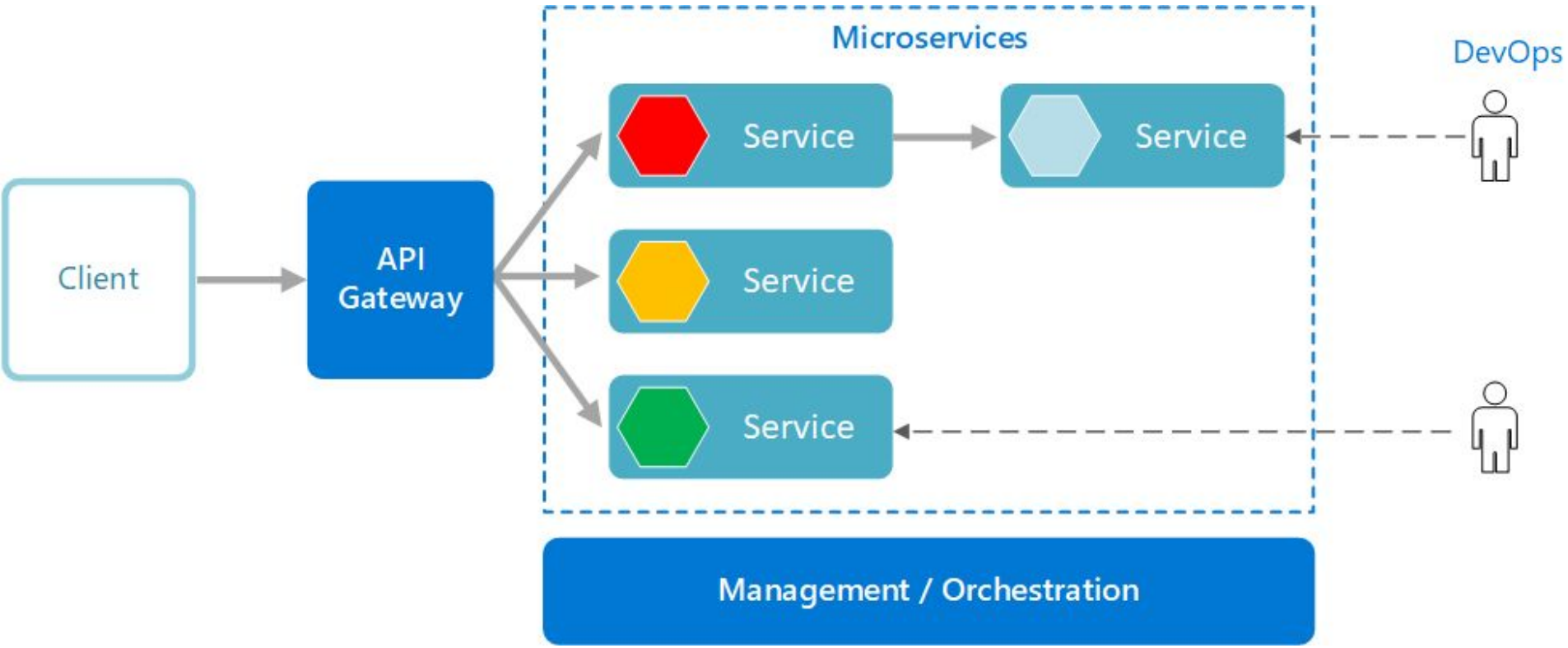

Admission
Transport
Staff Management
LMS
Assessment
Tablets
MIS
SMS
Fee
Fee Collection
Hostel
Registration
Payroll
Time Table
Front Office


PURCHASING
CUSTOMER WEB PORTAL
CRM & SALES
ERP
ENTERPRISE RESOURCE PLANNING
MANUFACTURING
DISTRIBUTION
FINANCE
DASHBOARDS
TIME & PROJECTS


UI
Business Logic
Data Access Layer
Monolithic Architecture
UI
Microservice
Microservice
Microservice
Microservice
Microservice
Microservice
Microservices Architecture

Micro Services

- Every app is an own service
- Own hardware
- Communicate via API / Message Queues / Service Brokers
  - Should not be cross database queries
- Horizontal and Vertical scaling JBHD
- Less risk in change
- Agile development


Microservices
DevOps
Client
API Gateway
Service
Service
Service
Service
Management / Orchestration

- Microservices are small, independent, and loosely coupled. A single small team of developers can write and maintain a service.
- Each service is a separate codebase, which can be managed by a small development team.
- Services can be deployed independently. A team can update an existing service without rebuilding and redeploying the entire application.
- Services are responsible for persisting their own data or external state. This differs from the traditional model, where a separate data layer handles data persistence.
- Services communicate with each other by using well-defined APIs. Internal implementation details of each service are hidden from other services.
- Supports polyglot programming. For example, services don't need to share the same technology stack, libraries, or frameworks.

Microservices architecture design - Azure Architecture Center | Microsoft Learn

Top 10 Microservices Patterns That Every Developer Should Know - GeeksforGeeks

Introduction to Microservices

Boris Scholl
Principal Program Manager –
Azure Service Fabric

Microsoft